

US009292964B2

(12) **United States Patent**  
**Silberstein et al.**

(10) **Patent No.:** **US 9,292,964 B2**  
(45) **Date of Patent:** **Mar. 22, 2016**

(54) **METHOD AND SYSTEM FOR ACCESSIBILITY AND CONTROL OF PARAMETERS IN SCENEGRAPHS**

(75) Inventors: **Ralph Andrew Silberstein**, Grass Valley, CA (US); **David Sahuc**, Nevada, CA (US); **Donald Johnson Childers**, Grass Valley, CA (US)

(73) Assignee: **GVBB Holdings S.A.R.L.**, Luxembourg (LU)

(\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 489 days.

(21) Appl. No.: **12/450,125**

(22) PCT Filed: **Jun. 28, 2007**

(86) PCT No.: **PCT/US2007/015149**

§ 371 (c)(1),  
(2), (4) Date: **Jun. 18, 2010**

(87) PCT Pub. No.: **WO2008/115196**

PCT Pub. Date: **Sep. 25, 2008**

(65) **Prior Publication Data**

US 2010/0271392 A1 Oct. 28, 2010

**Related U.S. Application Data**

(60) Provisional application No. 60/918,131, filed on Mar. 15, 2007.

(51) **Int. Cl.**  
**G06T 17/00** (2006.01)

(52) **U.S. Cl.**  
CPC ..... **G06T 17/00** (2013.01); **G06T 2210/61** (2013.01)

(58) **Field of Classification Search**

CPC ... G06T 17/005; G06T 2210/61; G06T 11/20; G06T 15/005; G06T 2200/24; G06T 13/00; G06T 15/00; G06T 17/00; G06T 2213/04;

G06T 19/00; G06T 17/10; G06T 19/20; G06T 2219/2016; G06T 11/001; G06T 2213/12; G06T 13/20; G06T 15/50; G06T 15/80; H04N 21/44012; H04N 21/4858; H04N 21/23412; H04N 21/47205; H04N 21/482; H04N 21/44222; H04N 21/84; H04N 21/8543; G06F 3/0482; G06F 9/4443; G06F 3/0484; G06F 8/34; G06F 8/38; A63F 2300/8082; A63F 13/10; A63F 2300/66; G01C 21/3638

USPC ..... 345/619, 650, 655  
See application file for complete search history.

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

5,982,388 A \* 11/1999 Nakagawa ..... 345/473  
6,426,745 B1 \* 7/2002 Isaacs et al. .... 345/419

(Continued)

**FOREIGN PATENT DOCUMENTS**

JP 2005-251161 9/2005  
WO WO2006108990 10/2006

**OTHER PUBLICATIONS**

Dietrich, A.; Wald, I.; Wagner, M.; Slusallek, P., "VRML scene graphs on an interactive ray tracing engine," Virtual Reality, 2004. Proceedings. IEEE, vol., no., pp. 109-282, Mar. 27-31, 2004.\*

(Continued)

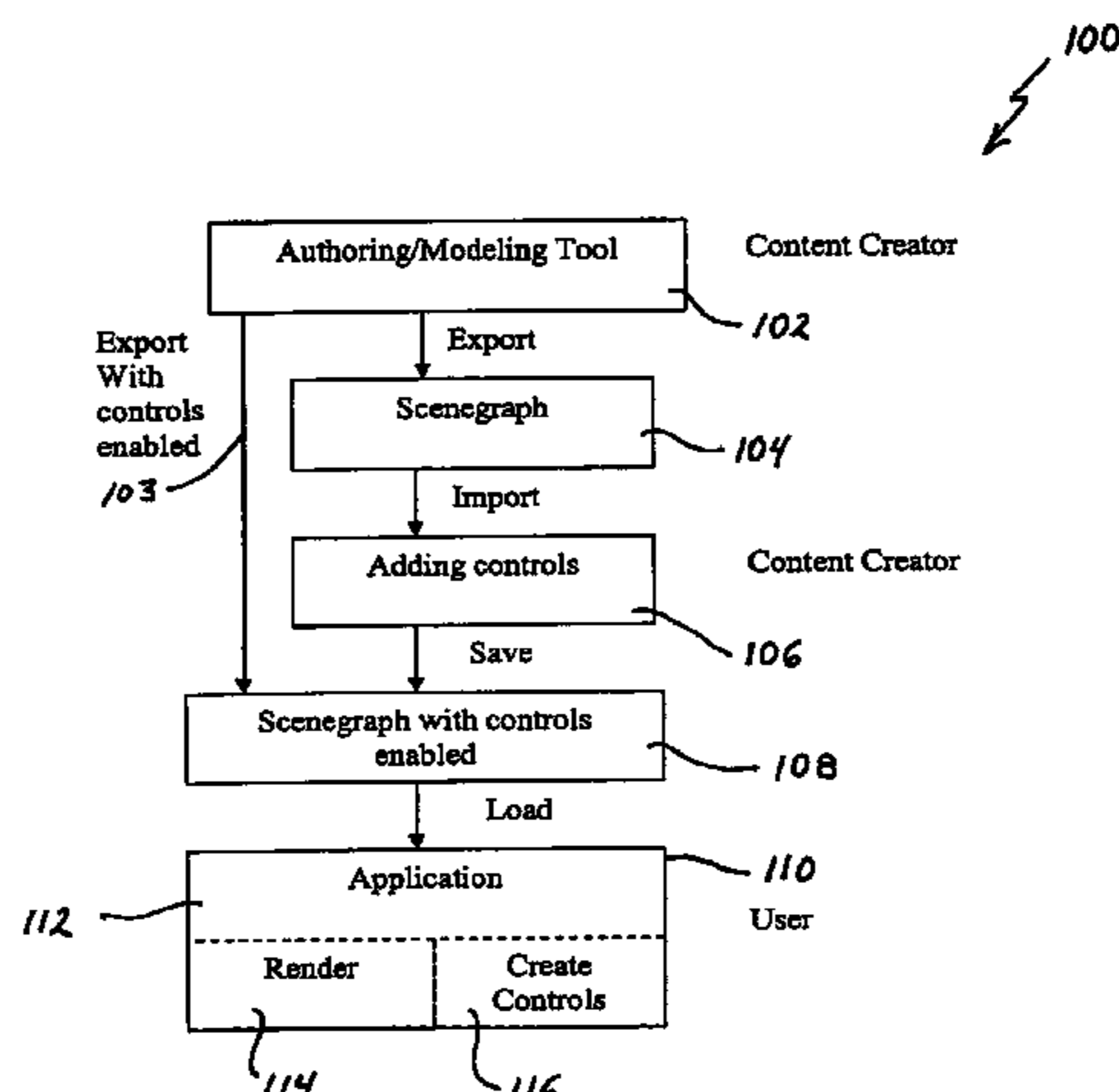
*Primary Examiner* — Haixia Du

(74) *Attorney, Agent, or Firm* — Arent Fox LLP

(57) **ABSTRACT**

A method and system for providing access to and control of parameters within a scenegraph includes redefining components or nodes' semantic within a scenegraph. The set of components or nodes (depending on the scenegraph structure) are required to enable access from the Application User Interface to selected scenegraph information. In one embodiment, a user interface is generated for controlling the scenegraph parameters. In addition, constraints can be implemented that allow or disallow access to certain scenegraph parameters and restrict their range of values.

**35 Claims, 5 Drawing Sheets**



(56)

**References Cited**

U.S. PATENT DOCUMENTS

6,693,645	B2 *	2/2004	Bourges-Sevenier	.....	345/619
7,030,890	B1 *	4/2006	Jouet et al.	.....	345/619
2003/0132937	A1 *	7/2003	Schneider et al.	.....	345/473
2004/0189667	A1 *	9/2004	Beda et al.	.....	345/619
2005/0035970	A1	2/2005	Wirtschaftler		
2005/0039176	A1 *	2/2005	Fournie	.....	717/156
2005/0131857	A1 *	6/2005	Fujiki et al.	.....	707/1

OTHER PUBLICATIONS

Johannes Behr, Patrick Dahne, and Marcus Roth. 2004. Utilizing X3D for immersive environments. In Proceedings of the ninth international conference on 3D Web technology (Web3D '04). ACM, New York, NY, USA, 71-78.\*

Akanksha et al., Animation toolkit based on a database approach for reusing motions and models, *Multimed Tools Appl* (2007) 32:293-327, Published online: Nov. 30, 2006.\*

First Office Action dated May 6, 2011 regarding China Application No. 200780052188.2.

Second Office Action dated Nov. 24, 2011 regarding China Application No. 200780052188.2.

Third Office Action dated Jun. 26, 2012 regarding China Application No. 200780052188.2.

Ken'ji Mase; "VR Rapid Prototyping Language InvenTcl, bit," Japan, Kyoritsu Shuppan Co., Ltd., Jan. 1, 2000, vol. 32, No. 1, pp. 36-41. IPRP dated Sep. 15, 2009 and Written Opinion dated Apr. 22, 2008 regarding PCT Application No. PCT/US2007/015149.

Notice of Reasons for Rejection dated Jan. 10, 2012 regarding Japan Patent Application No. JP2009-553557.

Muller A., et al.: "Multifaceted Interaction with a virtual engineering environment using a Scenegraph-oriented Approach" *Journal of WSCG, Univ. West Bohemia Czech Republic*, 2003, pp. 101-104, XP002475028.

Dietrich A., et al., "VRML Scene Graphs on an Interactive Ray Tracing Engine", *IEEE Virtual Reality*, Mar. 27-31, 2004, Chicago, IL

Behr J., et al.: "Utilizing X3D for Immersive Environments" *Proceedings of the Ninth International Conference on 3D Web Technology*, ISBN:1-58113-845-8, 2004, pp. 71-78, 182, XP002475027.

International Search Report dated Apr. 22, 2008.

\* cited by examiner

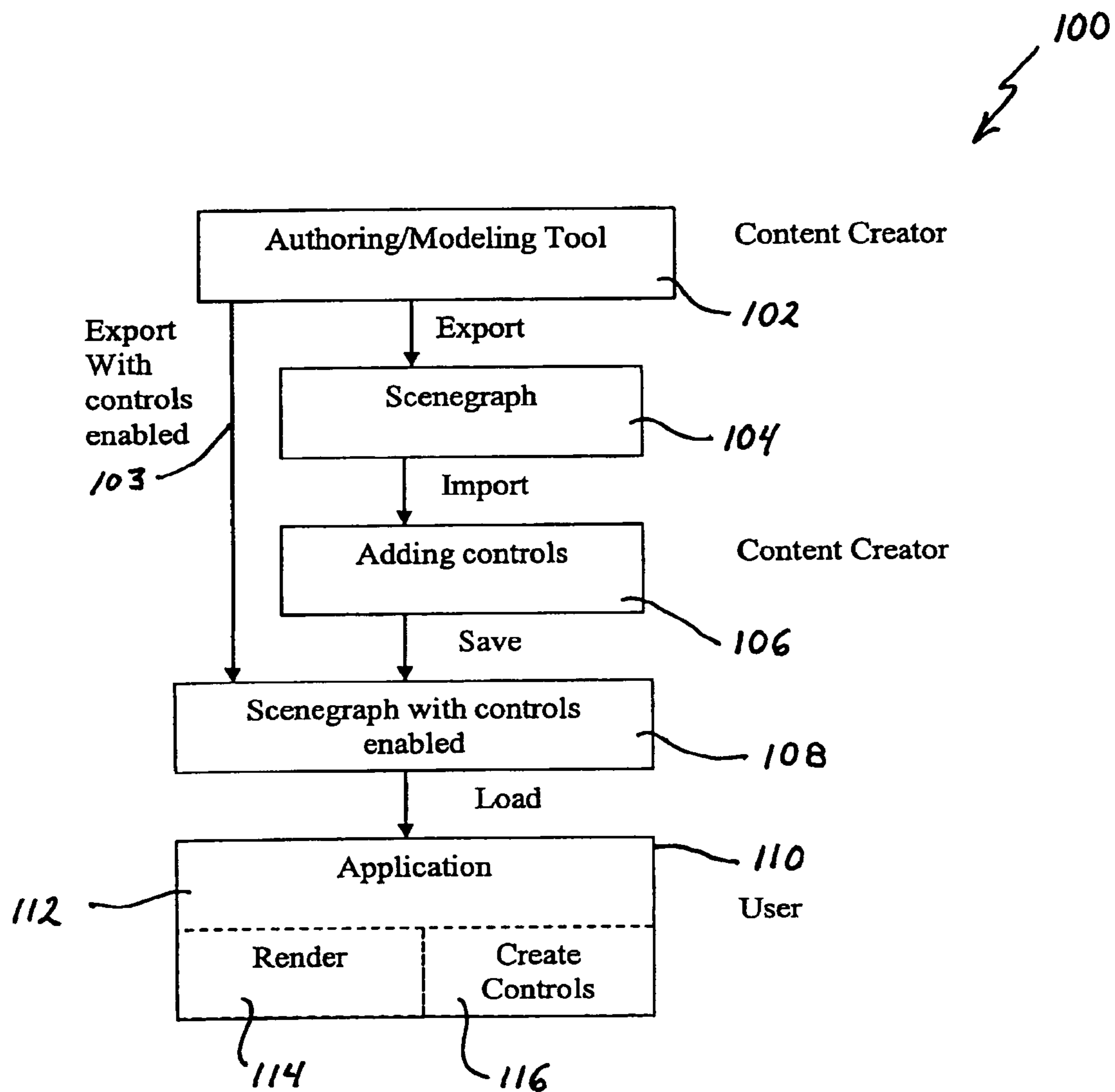


FIG. 1

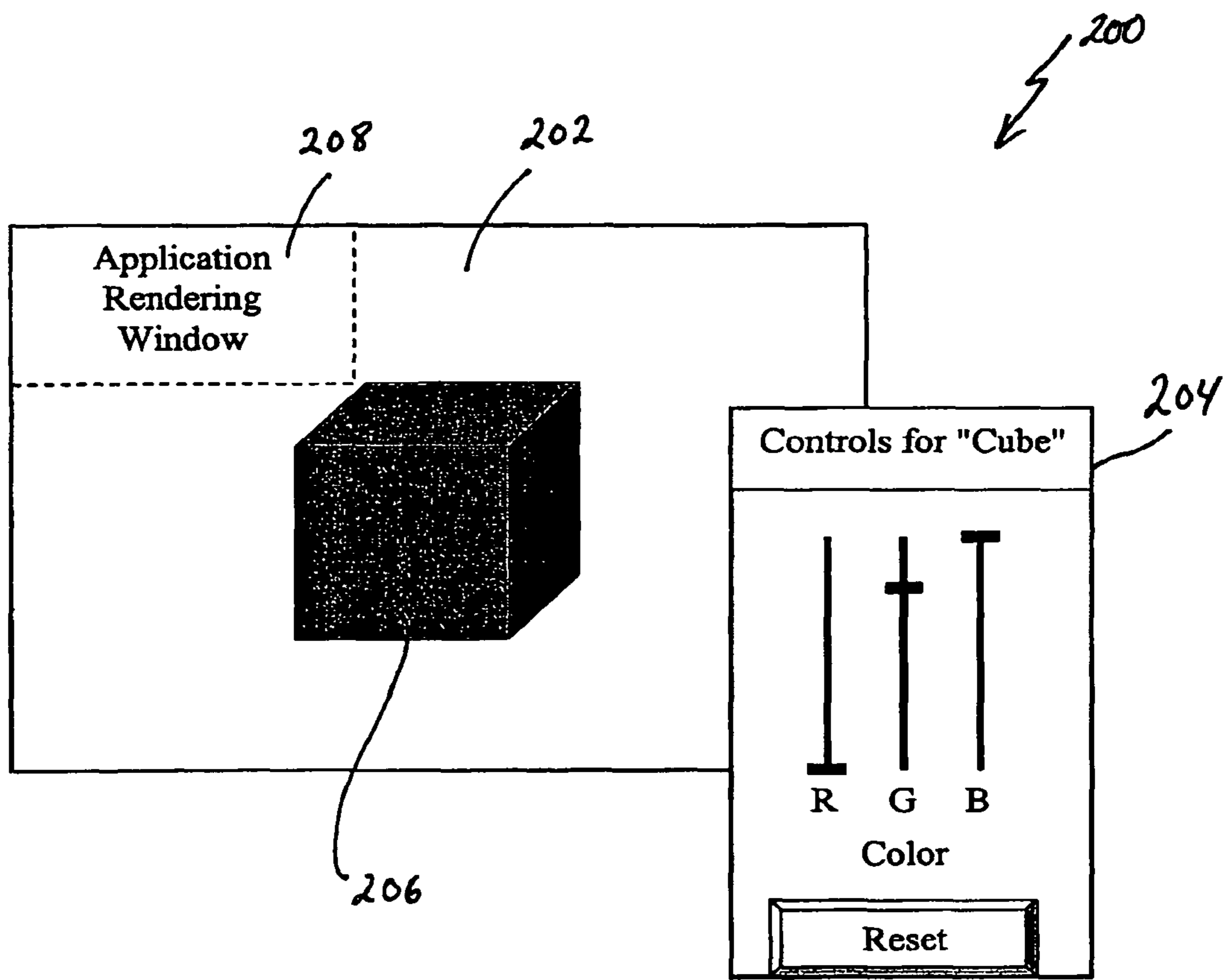


FIG. 2

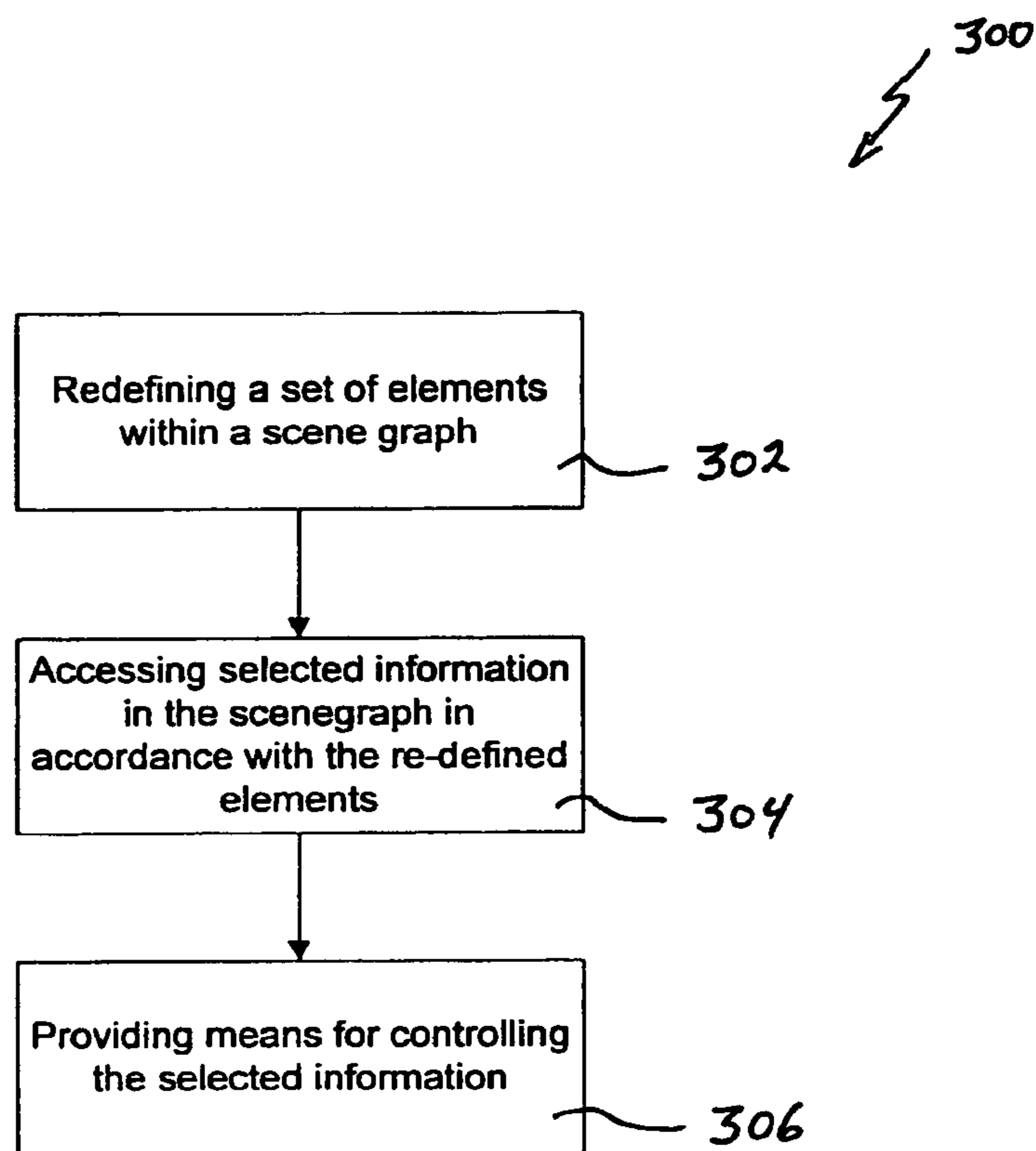


FIG. 3

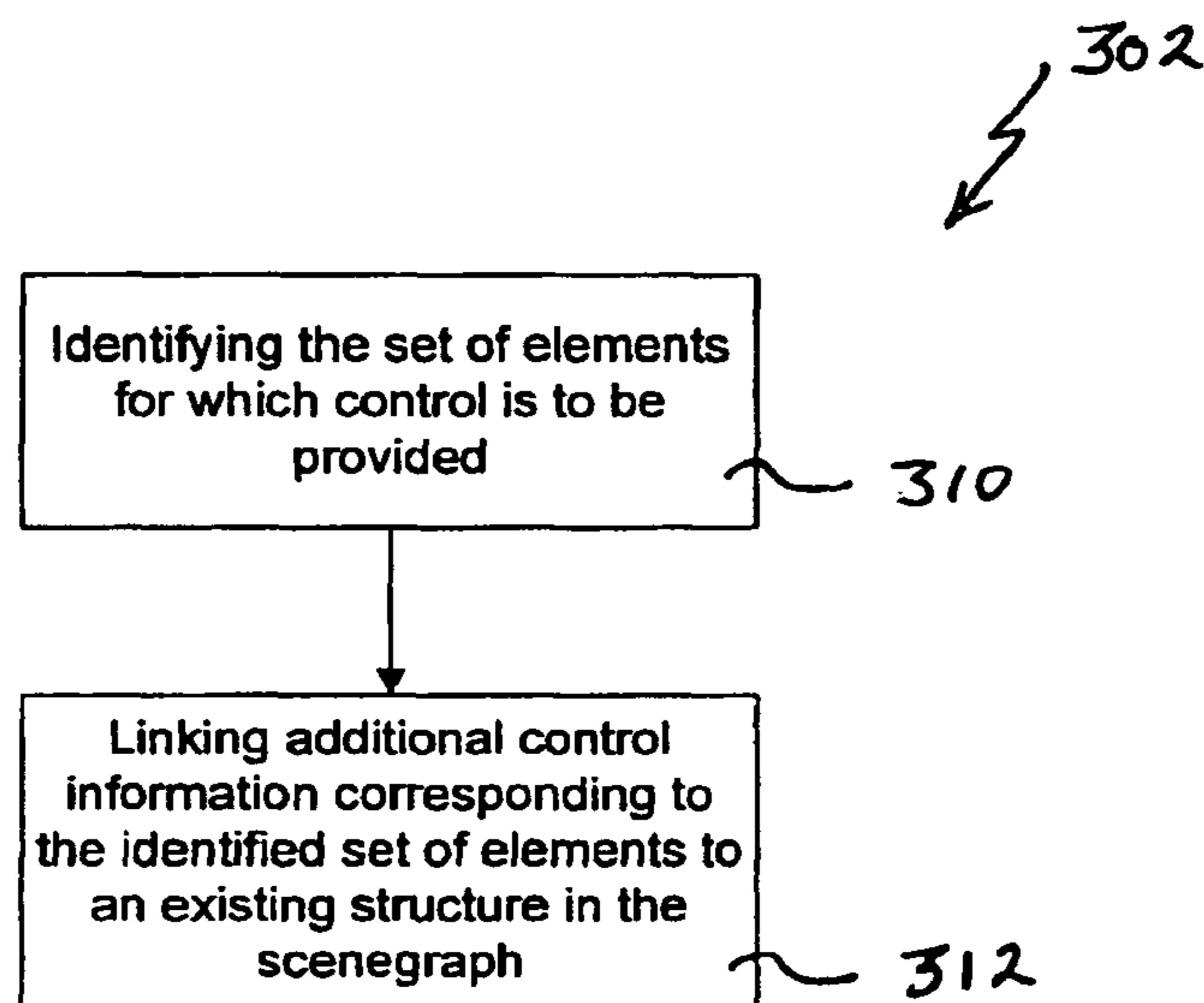


FIG. 4

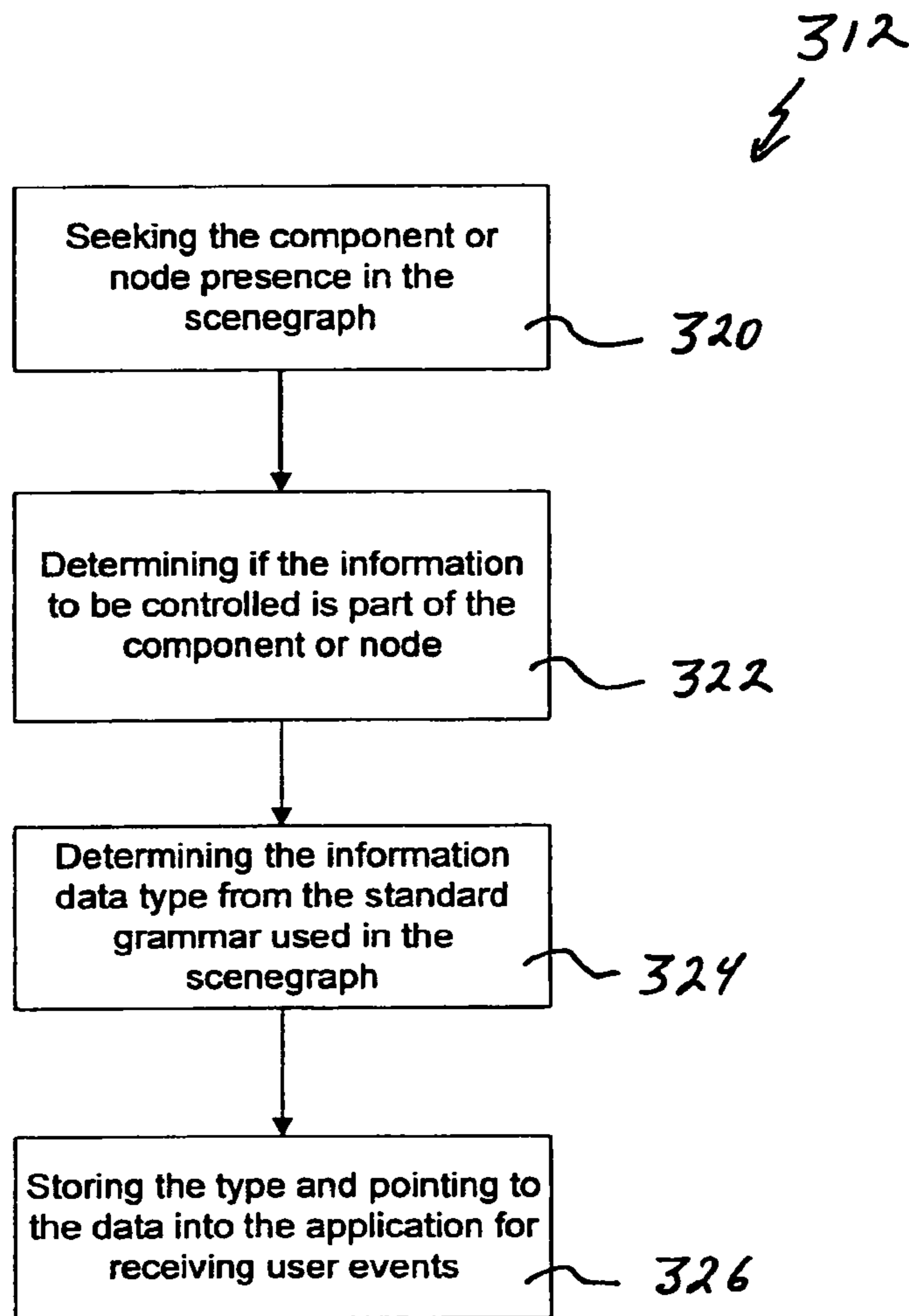


FIG. 5

## 1

**METHOD AND SYSTEM FOR  
ACCESSIBILITY AND CONTROL OF  
PARAMETERS IN SCENEGRAPHS**

This application claims the benefit, under 35 U.S.C. §365 of International Application PCT/US2007/015149 filed Jun. 28, 2007, which was published in accordance with PCT Article 21(2) on Sep. 25, 2008 in English and which claims the benefit of U.S. provisional patent application No. 60/918,131 filed Mar. 15, 2007.

TECHNICAL FIELD

The present invention relates to scenegraphs and more particularly to the accessibility and control of parameters during the rendering of a scenegraph.

BACKGROUND OF THE INVENTION

A scenegraph is a hierarchical structure composed of nodes, also called Objects, which can be either grouping nodes (intersection between branches) or leaves. Such structure is usually used as storage for 2D or 3D Graphics information (e.g., Geometrical Model) and their composition in order to render them.

A scenegraph can include animation information formed by either temporal interpolation means or frame based animation (e.g., in the case of Macromedia Flash). It can also contain sensors representing allowed User Interactions with the scene or objects composed within. Such animation must be fully described and interaction triggers routed correctly within the scenegraph in order to operate correctly.

The use of Scenegraphs in the Computer Graphics domain has been generalized for several years. Most 3D Authoring Tools (e.g., Autodesk 3D Studio or Maya) and 3D Rendering engines (e.g., Pixar RenderMan) internally (i.e., inside the application) use scenegraphs as data structures and as file formats for interchange (e.g., VRML, COLLADA). User controls during the rendering of a scenegraph are relatively limited, due to the fact their description must be embedded in the scenegraph. That is, unfortunately, there is no easy means in currently available technologies for defining direct Application User Interface controls for animations and parameters within a scenegraph or during rendering of the same.

SUMMARY OF THE INVENTION

Embodiments of the present invention provide an easy means to enable control of the parameters of a scenegraph and a method to treat those controls and present them to the user at the application level. Embodiments of the present invention also enable the definition of constraints for the user interaction, using value ranges, increments, defaults, and user or platform profiles for allowing or not allowing certain controls.

In one embodiment of the present invention, a method for controlling a scenegraph includes redefining a set of elements within a scene graph, accessing selected information in the scenegraph in accordance with the redefined elements, and providing means for controlling the selected information.

In an alternate embodiment of the present invention, a system for controlling a scenegraph includes a means for redefining a set of elements within a scenegraph, a means for accessing selected information in the scenegraph in accordance with the redefined elements, and a means for controlling the selected information

## 2

BRIEF DESCRIPTION OF THE DRAWINGS

The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 depicts a flow diagram of a process for enabling the control of scenegraph parameters in accordance with an embodiment of the present invention;

FIG. 2 depicts a graphical representation of a scenegraph and corresponding controls in accordance with an embodiment of the present invention;

FIG. 3 depicts a flow diagram of a method for accessibility and control of parameters in a scenegraph in accordance with an embodiment of the present invention;

FIG. 4 depicts a flow diagram of the re-defining step of the method of FIG. 3 in accordance with an embodiment of the present invention; and

FIG. 5 depicts a flow diagram of the linking step of the method of FIG. 4 in accordance with an embodiment of the present invention.

It should be understood that the drawings are for purposes of illustrating the concepts of the invention and are not necessarily the only possible configuration for illustrating the invention. To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION OF THE INVENTION

Many standards for a scenegraph description have been defined including VRML (Virtual Reality Modeling Language), X3D (eXtensible 3D), MPEG-4 Systems, COLLADA and Java3D. Such standards use either a textual representation or XML. In addition, several proprietary standards also exist (e.g., Microsoft DirectX format, NVIDIA NVSG, etc.). In all of these representations however, the primary means for enabling user interaction is to either integrate all interactions (sensors) and behaviors within the scenegraph to embed the whole model in a super-structure (e.g. prototype in VRML/X3D), or to develop a specific application using the scenegraph Application Interface (referred to as EAI-External Application Interface—in VRML, SAI-Scene Access Interface—in X3D, DirectX is the interface for the Microsoft format and MPEG-J is the interface in MPEG-4 Systems).

Describing the entire interaction and control within the scenegraph, as in the first listed solution, requires a rendering engine for the scenegraph to include an Execution engine to process the animations and interactions events (see description in the VRML handbook). This is because the Application User Interface in these cases does not have knowledge of the scenegraph. As such, specific source code must be developed in order to extract data from the Scenegraph and in order to give a user access to them. The second solution described above includes using a prototype as defined in VRML/X3D (see VRML handbook). This solution, as described in the VRML handbook is “A protection mechanism that allows an author to limit what can be done to an object”. The problem, however, is that this “prototype node” technique does not allow: 1) profiles management; 2) easy limitation of values for parameters (the only possible way is to use scripts in this case); and 3) it does not provide for any easy containment for creating a User Interface to control fields (i.e., there is no real name given to the parameters besides an identifier name for processing events in the Execution engine, and the prototype technique only helps in interactions between objects (nodes) within the scenegraph, not the user). In addition, using this



technique does not allow for the simple addition of new parameters to be presented to the user since the “prototype node” hides the actual scenegraph structure from the user.

The third technique described above uses a Scenegraph Access Interface (SAI) such as X3D specifications for the SAI. Such Interfaces are either bound to a language, (i.e., VRML EAI is a Java binding). The technique here is then to develop a specific User Interface to get access to the scenegraph information. This technique allows access to any member of the scenegraph. However, accessing all the fields means the user must know the whole scenegraph structure and have knowledge of all its structural elements (nodes, fields, etc). Moreover, names in the scenegraph might not be clear enough to a user, due to the name of their grammatical composition, e.g. the space character is not allowed.

Embodiments of the present invention can apply to any existing scenegraph description language (e.g., textual or XML), provided the implementation matches the chosen scenegraph file format grammar. A parser can then decide to treat or not treat the nodes defined by the present principles, allowing or not allowing the feature of the present invention. The application can also choose not to display the User Interface controls defined by the present principles. For example, not treating the new feature of the present invention would be very straight forward on the parsing side, as all nodes have bounds, sometimes represented as brackets (e.g., ‘{’ and ‘}’) or XML tag brackets (e.g., ‘<node>’ and ‘</node>’). The parser then only needs to ignore the complete feature declaration in the scenegraph, limited to its brackets.

Embodiments of the present invention include a set of additional components or nodes (depending on the scenegraph structure) needed to enable the access to the selected scenegraph information from the Application User Interface without modifying the original scenegraph description. The new components or nodes enable the redefining of the targeted scenegraph information’s semantic to be presentable to the user, for example, with a comprehensible name and modifier on the user interface. This information can then be controlled through a newly created Graphical Interface widgets provided to the user through their graphical user interface.

For example, in one embodiment of the present invention, Feature Identifiers can be represented with matching closing and opening brackets, defined in compliance with the scenegraph grammar. In addition, Profiles Identifiers can be represented by a list of strings or identifiers describing the profiles. A profile can include a user, for example, describing access rights, or platform, for example, describing capabilities and/or profile. Containment of those profile identifier strings can be application dependant, or either a name description or a numeric value. If this list of strings is empty, all defined controls will be available to all users or platforms. Certain implementations of alternate embodiments of the present invention, however, may not use this information at all (i.e., instances in which profiles are not present on a platform, such like a decoder STB—Set Top Box—), while other embodiments can manage profiles at a separate layer in the application (i.e., in the case of static profiles in a platform, like a switcher). Embodiments of the present invention can further include a List of controlled items in the scenegraph for each of them.

The identifiers in the scenegraph include a list of strings or identifiers which define the targeted information in the scenegraph and are usually composed of the node’s name definition or identifier and the node’s field’s name or identifier to be controlled. If an identifier is not found at the current parsing level in the scenegraph, its control should not be activated. The application can then retrieve the field value type (e.g.,

float, integer) as a result of its definition in the scenegraph. This field is essentially a list of strings that allow grouped modification, for example, several color fields at the same time, under a single User Interface component. It is therefore necessary that the fields listed are of the same type.

Embodiments of the present invention can further include a name appearing on the application User Interface. This will usually be different from the identifier name mentioned in the previous paragraph, as usually identifiers in a scenegraph are syntactically restricted in the grammar (no space or specific character) and may not present an appropriate description (i.e., these identifiers are usually automatically generated by the graphics modeling tool).

In accordance with embodiments of the present invention, a reset or factory value can include a value to set the parameter when a “reset” or “factory default” button is activated on the User Interface. This value may be different from the value in the scenegraph itself, since the scenegraph stores values at a specific instant. Even further, minimum and maximum values can be included which define a range of values valid for the parameter. The minimum and maximum values define limits for the User Interface controls.

The Application of embodiments of the present invention can further include a Step Increment, which corresponds to a stepwise increment (absolute value) to apply to a parameter (positively or negatively) on the User Interface by, for example, pushing a ‘+’ or ‘-’ button, and defined Profiles for which this control is enabled. That is, selected profiles identifiers (e.g., those listed above or other application defined identifiers) for which this control is enabled can be listed here.

In one embodiment of the present invention, all values are represented in the field type numeric domain. If the chosen scenegraph does not provide a generic field type, for representing, for example, Floating values and Booleans, the largest domain can be used to define those parameters. For example, a float representation can be used and the value can be cast as an integer or boolean. In accordance with embodiments of the present invention, all values can be an array of numbers, as many parameters can be composed of more than one element or component, for example, a color with RGB values (Red, Green and Blue). If the number of elements/components making up the controlled field and the number of elements/components contained in the minimum, maximum, reset and increment values differ, only the minimum number of elements on those latter values matching the number of elements/components in the controlled field shall be used. For example, if the controlled field represents a RGB color with 3 components and the “reset” value is defined as “[1.0 1.0 1.0 1.0]”, the system will only use the 3 first elements and leave the last one unused.

If the node in which the controlled field is contained is destroyed during the execution of the scenegraph, the User Interface control linked to this node can be deactivated or destroyed as well. Embodiments of the present invention also allow controlled items to be grouped. This allows for the grouping of a list of parameters on the User Interface in a more natural way, for example, a “car” composed of “wheels”, or allows for an override of the scenegraph natural hierarchical structure for presenting the information to the user. This grouping feature can be named as well, and can recursively contain other groups.

Embodiments of the present invention can be implemented on any existing scenegraphs, with the only restrictions being those inherent to each scenegraph model. For example, VRML 2.0 does not provide a generic field type, thus only “SFFloat” can be used as the largest domain.

## 5

For example, a VRML 2.0 nodes implementation of the present invention can be represented as follows:

---

```
ControlProfiles {
  field MFString profiles [ ]
}
```

---

The “ControlProfiles” nodes can be discarded if profiles are defined at the application layer.

---

```
ControlGroup {
  field SFString name “car”
  field MFNode items [ ]
  field MFString profiles [ ]
}
```

---

The “items” field can contain a list of “ControlItems” (see below) or other “ControlGroup” nodes. The “name” field defines the group name to be presented to the user, for example, “car”. The “profiles” field as presented overrides the profile designation on the individual items in the “items” list, allowing the profile to easily control the whole group.

---

```
ControlItem {
  field MFString identifiers [ ]
  field SFString name “”
  field MFFloat resetValue [ ]
  field MFFloat minValue [ ]
  field MFFloat maxValue [ ]
  field MFFloat stepIncrement [ ]
  field MFString profiles [ ]
}
```

---

In the embodiment described above, the “profiles” list contains a list of profiles as defined by the “ControlProfiles” node or application-dependant values.

In an alternate embodiment of the present invention, a PROTO structure concept within VRML 2.0 implementation can be represented as follows:

---

```
CONTROLGROUP [
  field MFString profiles [ ]
  field fieldtypename “name” IS [identifiers]
  [resetValue] [minValue] [maxValue]
  [stepIncrement] [profiles]
]
{
  # Internal Scenegraph section
}
```

---

In yet an alternate embodiment of the present invention, an X3D nodes implementation of the present invention can be represented as follows:

---

```
<ControlGroup name=“” profiles=[ ]>
  <ControlItem identifiers=[ ] name= “” resetValue=[ ]
  minValue=[ ] maxValue=[ ] stepIncrement=[ ]
  profiles=[ ]/>
</ControlGroup>
```

---

FIG. 1 depicts a flow diagram of a process for enabling the control of scenegraph parameters in accordance with an embodiment of the present invention. The process 100 of FIG. 1 begins with an Authoring or Modeling Tool 102 including the control methods of an above described embodiment the present invention. At step 103, a content creator exports a

## 6

subject scenegraph. If the scenegraph has the controls enabled, the process continues to step 108. If the authoring/modeling tool does not supply such functionality to the scenegraph a specific module can be created to add the controls afterwards. For example, when the authoring modeling tool 102 does not supply such functionality, an author exports the scenegraph 104, to module 106 for adding the necessary controls. The scenegraph with the controls is then enabled 108 and loaded into the application block 110. The Application 112 then loads the scenegraph to render it 114 and can now create the controls 116 (and make them available to the user) on the Graphical User Interface.

In one embodiment of the present invention, the application 112 can directly embed the create controls module 116 to add the user controls. This can be useful for a content creator to verify correct execution and rendering of the scenegraph and its corresponding controls.

FIG. 2 depicts a graphical representation of a scenegraph and corresponding controls in accordance with an embodiment of the present invention. The embodiment of FIG. 2 shows a rendering of a scenegraph 200 (simple cube in this case) and its controls (simple RGB color selection in this case) by an application which supports the methods of the present invention. The scenegraph includes a white background 202, a light-blue cube 206 displayed on the application rendering window 208 and the controls window 204 created by the application with respect to the interpretation of the VRML nodes embodiment. The scenegraph example used in FIG. 2 is written using VRML 2.0 Scenegraph, with the profiles being deactivated. In accordance with one embodiment of the present invention, the VRML 2.0 description of the scenegraph can be represented as follows:

---

```
Shape {
  geometry Cube { }
  appearance Appearance {
    material DEF MAT Material {
    }
  }
}
ControlGroup {
  name “Cube”
  items ControlItem {
    identifiers “MAT.diffuseColor”
    name “Color”
    resetValue [ 0.0 0.0 0.0 ]
    minValue [ 0.0 0.0 0.0 ]
    maxValue [ 1.0 1.0 1.0 ]
    stepIncrement [ 0.1 0.1 0.1 ]
    profiles [“”]
  }
}
```

---

As depicted in the example above, the application 112 uses the Control Item to establish a binding between the name “Color” from the Control Group “Cube” and the scenegraph field “diffuseColor”, part of the “MAT” node, which type is SFCOLOR in the VRML 2.0 specifications, and thus functionally supports the desired controls (i.e., 3 editable fields for Red, Green, and Blue color levels). This is shown by the controls window 204 in FIG. 2 where the individual R, G and B controls are shown.

FIG. 3 depicts a flow diagram of a method for accessibility and control of parameters in a scenegraph in accordance with an embodiment of the present invention. The method 300 of FIG. 3 begins at step 302 in which a set of elements within the scenegraph are initially redefined. The method then proceeds to step 304.

At step 304, once redefined the selected information in the scenegraph is accessed in accordance with the redefined elements. The method then proceeds to step 306.

At step 306, when the selected information has been accessed, means are provided to control the selected information.

FIG. 4 depicts a flow diagram of the re-defining step 302 of the method 300 of FIG. 3 in accordance with an embodiment of the present invention. In order to redefine the elements as defined in the method 300 of FIG. 3, initially the set of elements for which control is to be provided is identified at step 310. At step 312, once identified, additional control information is linked to the corresponding identified set of elements to the existing structure in the scenegraph.

FIG. 5 depicts a flow diagram of the linking step 312 of the method of FIG. 4 in accordance with an embodiment of the present invention. In the embodiment of FIG. 5, in order to link the additional information, there is an initial seeking of the component or node presence within the scenegraph at step 320. At step 322, it is then determined if the information to be controlled is part of the component or node. Once this determination has been made, another determination of the information data type from the standard grammar used in the scenegraph is made at step 324. At step 325, with all the above described determinations made, the type and point to the data can be stored into the application for receiving user events.

The various aspects, implementations, and features may be implemented in one or more of a variety of manners, even if described above without reference to a particular manner or using only one manner. For example, the various aspects, implementations, and features may be implemented using, for example, one or more of a method, an apparatus, an apparatus or processing device for performing a method, a program or other set of instructions, an apparatus that includes a program or a set of instructions, and a computer readable medium.

An apparatus may include, for example, discrete or integrated hardware, firmware, and software. As an example, an apparatus may include, for example, a processor, which refers to processing devices in general, including, for example, a microprocessor, an integrated circuit, or a programmable logic device. As another example, an apparatus may include one or more computer readable media having instructions for carrying out one or more processes.

A computer readable medium may include, for example, a software carrier or other storage device such as, for example, a hard disk, a compact diskette, a random access memory ("RAM"), or a read-only memory ("ROM"). A computer readable medium also may include, for example, formatted electromagnetic waves encoding or transmitting instructions. Instructions may be, for example, in hardware, firmware, software, or in an electromagnetic wave. Instructions may be found in, for example, an operating system, a separate application, or a combination of the two. A processor may be characterized, therefore, as, for example, both a device configured to carry out a process and a device that includes a computer readable medium having instructions for carrying out a process.

Having described preferred embodiments for a method and system for providing access to and control of parameters within a scenegraph (which are intended to be illustrative and not limiting), it is noted that modifications and variations can be made by persons skilled in the art in light of the above teachings. It is therefore to be understood that changes may be made in the particular embodiments of the invention disclosed which are within the scope and spirit of the invention as outlined by the appended claims. While the forgoing is

directed to various embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof.

The invention claimed is:

1. A method for controlling a scenegraph during rendering comprising:
  - identifying a set of elements within a scenegraph having an original scenegraph description, the set of elements being those for which additional control is to be provided;
  - linking the additional control information, during rendering, to existing scenegraph structure;
  - accessing selected information in the scenegraph using a processor in accordance with the elements;
  - during rendering, providing an interlace for controlling the selected information including:
    - providing identifiers for at least one of the elements for enabling a user to identify the elements to be controlled, and
    - providing additional components that enable access to the elements to be controlled without modifying the original scenegraph description, wherein user control for the elements is not embedded in the original description of the scenegraph; and
    - constraining activation of control for a feature, wherein constraint is based on a feature identifier for the feature.
2. The method of claim 1, wherein said identified set of elements comprises at least one component or at least one node depending on the scenegraph's structure.
3. The method of claim 1, wherein said providing means for controlling comprises providing a user interface with controls relating to the accessed selected information.
4. The method of claim 1, wherein said providing further comprises adding one or more additional components or nodes to enable access to the selected scenegraph information from an application user interface.
5. The method of claim 1, wherein said providing means for controlling comprises defining at least one constraint for user interaction with the scenegraph.
6. The method of claim 5, wherein said defining comprises using a set of elements comprising at least one of value ranges, increments and user or platform profiles.
7. The method of claim 1, comprising:
  - determining whether the scenegraph provides a generic field type; and
  - using a largest domain in the scenegraph for said identifying of elements when the scenegraph does not provide a generic field type.
8. The method of claim 1, wherein said linking additional information comprises:
  - seeking the component or node presence in the scenegraph;
  - determining if the information to be controlled is part of the component or node;
  - determining the information data type from the scenegraph; and
  - storing the data type and pointing to the data into application for receiving the user events, wherein the data type is stored on a non-transitory computer readable information storage medium.
9. The method of claim 1, wherein said identifying accessing and providing are performed during scenegraph authoring using a scenegraph authoring tool.
10. The method of claim 1, wherein said identifying, accessing and providing are performed after authoring of a scenegraph is completed.

11. The method of claim 10, further comprising:  
exporting a scenegraph. from an authoring tool;  
importing controls into the exported scenegraph.;  
saving the scenegraph with controls enabled; and  
loading the scenegraph with enabled controls into an appli- 5  
cation.

12. The method of claim 1, wherein the set of elements for  
which additional control is to be provided are grouped using  
a parameter.

13. The method of claim 1, wherein additional control for 10  
parameter groups is limited to a range of values valid for the  
parameter.

14. The method of claim 1, wherein the user control is  
provided through a graphical interface widget provided to the 15  
user through the user interface.

15. The method of claim 1, wherein constraint is based on  
a profile identifier.

16. The method of claim 1, wherein when a previously  
created scenegraph does not include user control to access the 20  
elements, the method includes:

exporting the scenegraph. to a module for adding the con-  
trols.

17. The method of claim 16, wherein the module links  
additional control information to an element in the existing 25  
structure of the scenegraph.

18. The method of claim 1, wherein control of the feature is  
constrained from activation when an identifier for the feature  
is not found at a parsing level in the scenegraph.

19. An apparatus for controlling a scenegraph during ren- 30  
dering comprising:

means for, during rendering, identifying a set of elements  
within a scenegraph having an original scenegraph  
description, the set of elements being those for which  
additional control is to be provided, and linking the 35  
additional control information to existing scenegraph  
structure;

means for accessing selected information in the scene-  
graph in accordance with the elements; and

means for controlling the selected information, wherein 40  
the means for controlling provides identifiers for at least  
one of the elements for enabling a user to identify the  
elements to be controlled and provides additional com-  
ponents that enable access to the elements to be con-  
trolled without modifying the original scenegraph 45  
description, wherein user control for the elements is not  
embedded in the original description of the scenegraph,  
wherein control is constrained for a feature based on a  
feature identifier for the feature.

20. The apparatus of claim 19, wherein said identified set of 50  
elements comprises at least one component or at least one  
node depending on the scenegraph's structure.

21. The apparatus of claim 19, wherein said means for  
controlling comprises a graphical user interface with controls  
relating to the accessed selected information.

22. The apparatus of claim 19, wherein said controlling 55  
means further comprises means for adding one or more addi-  
tional nodes to enable access to the selected scenegraph infor-  
mation from an application user interface.

23. The apparatus of claim 19, wherein the set of elements  
for which additional control is to be provided are grouped  
using a parameter.

24. The apparatus of claim 19, wherein additional control  
for parameter groups is limited to a range of values valid for  
the parameter.

25. The apparatus of claim 19, wherein linking additional  
formation comprises:

seeking the component or node presence in the scenegraph;  
determining if the information to be controlled is part of the  
component or node;

determining the information data type from the scene-  
graph; and

storing the data type and pointing to the data into applica-  
tion for receiving the user events, wherein the data type  
is stored on a non-transitory computer readable infor-  
mation storage medium.

26. The apparatus of claim 19, wherein the means for  
controlling exports a scenegraph from an authoring tool;

imports controls into the exported scenegraph;

saves the scenegraph with controls enabled; and

loads the scenegraph with enabled controls into an appli-  
cation.

27. The apparatus of claim 19, wherein constraint is based  
on a profile identifier.

28. The apparatus of claim 19, wherein when a previously  
created scenegraph does not include user control to access the  
elements, the means for controlling exports the scenegraph to  
a module for adding the controls, and

wherein the module links additional control information to  
an element in the existing structure of the scenegraph.

29. The apparatus of claim 19, wherein control of the  
feature is constrained from activation when an identifier for  
the feature is not found at a parsing level in the scenegraph.

30. The apparatus of claim 19, wherein said means for  
controlling the selected information defines at least one con-  
straint for user interaction with the scenegraph.

31. The apparatus of claim 30, wherein said defining com-  
prises using a set of elements comprising at least one of value  
ranges, increments and user or platform profiles.

32. The apparatus of claim 19, further comprising:  
means for determining whether the scenegraph provides a  
generic field type, wherein the means for identifying use  
a largest domain in the scenegraph for said identifying of  
elements when the scenegraph does not provide a  
generic field type.

33. The apparatus of claim 19, wherein said means for  
identifying, means for accessing and means for controlling  
are configured to perform the identifying, accessing, and  
controlling during scenegraph authoring using a scenegraph  
authoring tool.

34. The apparatus of claim 19, wherein said means for  
identifying, means for accessing and means for controlling  
are configured to perform the identifying, accessing, and  
controlling after authoring of a scenegraph is completed.

35. The apparatus of claim 19, wherein the means for  
controlling provides user control through a graphical inter-  
face widget provided to the user through the user interface.

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 9,292,964 B2  
APPLICATION NO. : 12/450125  
DATED : March 22, 2016  
INVENTOR(S) : Ralph Andrew Silberstein et al.

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page:

The first or sole Notice should read --

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b)  
by 496 days.

Signed and Sealed this  
Second Day of August, 2016



Michelle K. Lee  
*Director of the United States Patent and Trademark Office*